Journal of Machine Learning Research xx (2012) pp Submitted 6/12; Published xx

Learning Mixed Graphical Models

Jason D Lee jdl17@stanford.edu
Institute of Computational and Mathematical Engineering
Stanford University
Stanford, CA 94305, USA

Trevor Hastie hastie@stanford.edu
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Editor: xx

Abstract

We consider the problem of learning the structure of a pairwise graphical model over continuous and discrete variables. We present a new pairwise model for graphical models with both continuous and discrete variables. The structure and parameters of this model are learned using the pseudo-likelihood approximation with group-sparsity regularization. The pairwise model is also extended to incorporate features. Two algorithms for solving the resulting optimization problem are presented. The proposed models are compared with competing methods on synthetic data and a survey dataset.

Keywords: mixed graphical models, Markov random field, l_1 regularization, structure learning

1. Introduction

Many authors have considered the problem of learning the edge structure and parameters of sparse undirected graphical models. We will focus on using the l_1 regularizer to promote sparsity. This line of work has taken two separate paths: one for learning continuous valued data and one for learning discrete valued data. In this work, we consider learning mixed models with both continuous variables and discrete variables.

For continuous models, previous work assumes that the model is a multivariate Gaussian (Gaussian graphical model) with mean 0 and inverse covariance Θ. Θ is then estimated Θ via the graphical lasso by minimizing the regularized negative log-likelihood $\ell(\Theta) + \lambda \left\|\Theta\right\|_1$. Several efficient methods for solving this can be found in (Friedman et al., 2008a; Banerjee et al., 2008). Because the graphical lasso problem is computationally challenging, several authors considered methods related to the pseudolikelihood (PL) and node-wise l_1 least squares (Meinshausen and Bühlmann, 2006; Friedman et al., 2010; Peng et al., 2009). For discrete models, previous work focuses on estimating a pairwise Markov random field of the form $p(y) \propto \exp \sum_{r \leq j} \phi_{rj}(y_r, y_j)$. The maximum likelihood problem is intractable for models with a moderate to large number of variables (high-dimensional) because its requires evaluating the partition function and its derivatives. Previous work considers the pseudolikelihood approach and the maximum likelihood approach (Schmidt, 2010; Schmidt et al., 2008; Höfling and Tibshirani, 2009; Jalali et al., 2011; Lee et al., 2006; Ravikumar et al., 2010).

Our main contribution is to propose a model that connects the discrete and continuous models. Our model in the case of discrete variables, is a pairwise Markov random field and in the case of continuous variables is a Gaussian graphical model. In Section 2, we introduce a new mixed graphical model and previous approaches to modeling mixed data. Section 3 discusses the likelihood and the pseudolikelihood approaches to parameter estimation. Section 4 discusses how to use the l_1 penalty for edge selection, Section 5 presents the pseudolikelihood for the mixed model and its relationship to generalized linear models and Section 6 presents empirical results on synthetic and real datasets.

2. Mixed Graphical Model

We propose a pairwise graphical model on continuous and discrete variables. The continuous variables follow a multivariate Gaussian and the discrete variables follow a pairwise Markov random field model

$$p(x,y;\Theta) \propto \exp\left(\sum_{s=1}^{p}\sum_{t=1}^{p} -\frac{1}{2}\beta_{st}x_s x_t + \sum_{s=1}^{p}\alpha_s x_s + \sum_{s=1}^{p}\sum_{j=1}^{q}\rho_{sj}(y_j)x_s + \sum_{j=1}^{q}\sum_{r=1}^{q}\phi_{rj}(y_r,y_j)\right) \tag{1}$$

for a graphical model with p continuous variables and q discrete variables. The joint model is parametrized by $\Theta = [\{\beta_{st}\},\{\alpha_s\},\{\rho_{sj}\},\{\phi_{rj}\}]^1$. The discrete y_r takes on L_r states. The model parameters are β_{st} continuous-continuous edge potential, α_s continuous node potential, $\rho_{sj}(y_j)$ continuous-discrete edge potential, and $\phi_{rj}(y_r,y_j)$ discrete-discrete edge potential. In the case of continuous variables only, we recover the familiar multivariate Gaussian parametrized by the symmetric positive-definite inverse covariance matrix $B = \{\beta_{st}\}$, a symmetric positive definite matrix. In the case of discrete variables only, we recover the pairwise Markov random field parametrized by edge potentials ϕ_{rj} and node potentials ϕ_{rr}.

The conditional distribution of the continuous variables given the discrete follow a multivariate gaussian distribution, $p(x|y) = \mathcal{N}(\mu(y), B^{-1})$. Each of these gaussian distributions share the same inverse covariance matrix B but differ in the mean parameter. By standard multivariate gaussian calculations,

$$p(x|y) = \mathcal{N}(B^{-1}\gamma(y), B^{-1}) \tag{2}$$

$$\{\gamma(y)\}_s = \sum_j \rho_{sj}(y_j) \tag{3}$$

$$p(y) \propto \exp\left(\sum_{j=1}^{q}\sum_{r=1}^{j}\phi_{rj}(y_r,y_j) + \frac{1}{2}\gamma(y)^T B^{-1}\gamma(y)\right) \tag{4}$$

[1] $\rho_{sj}(y_j)$ is a function taking L_j values $\rho_{sj}(1),\ldots,\rho_{sj}(L_j)$. Similarly, $\phi_{rj}(y_r,y_j)$ is a bivariate function taking on $L_r \times L_j$ values. Later, we will think of $\rho_{sj}(y_j)$ as a vector of length L_j and $\phi_{rj}(y_r,y_j)$ as a matrix of size $L_r \times L_j$.

Thus we see that the continuous variables conditioned on the discrete are multivariate gaussian with common covariance, but with means that depend on the value of the discrete variables. The means depend additively on the values of the discrete variables since $\{\gamma(y)\}_s = \sum_{j=1}^r \rho_{sj}(y_j)$. The marginal $p(y)$ has a known form, so for models with few number of discrete variables we can sample efficiently. The single conditionals $p(x_s|x_{\backslash s}, y)$ is a gaussian linear regression and $p(y_r|x, y_{\backslash r})$ is a (multiclass) logistic regression. We discuss this further in Section 3.1.

2.1 Previous work on mixed graphical models

In Lauritzen (1996), a type of mixed graphical model is proposed. This model has the property that conditioned on discrete variables, $p(x|y) = \mathcal{N}(\mu(y), \Sigma(y))$. The homogeneous mixed graphical model enforces common covariance, $\Sigma(y) \equiv \Sigma$. Thus our proposed model is a special case of Lauritzen's mixed model with the following assumptions: common covariance, additive mean assumptions and the marginal $p(y)$ factorizes as a pairwise discrete Markov random field. With these two assumptions, the full model simplifies to the pairwise model presented. Although the full model is more expressive, the number of parameters scales exponentially with the number of discrete variables. For each state of the discrete variables there is a mean and covariance. Consider an example with q binary variables and p continuous variables; the full model requires estimates of 2^q mean vectors and covariance matrices in p dimensions. Even if the homogeneous constraint is imposed on Lauritzen's model, there are still 2^q mean vectors for the case of binary discrete variables. In comparison, the pairwise model has number of parameters $O((p+q)^2)$.

3. Parameter Estimation: Maximum Likelihood and Pseudolikelihood

Given samples $(x_i, y_i)_{i=1}^n$, we want to find the maximum likelihood estimate of Θ. This can be done by minimizing the negative log-likelihood of the samples:

$$\ell(\Theta) = -\sum_{i=1}^n \log p(x_i, y_i; \Theta) \text{ where} \tag{5}$$

$$\begin{aligned}\log p(x, y; \Theta) = &\sum_{s=1}^p \sum_{t=1}^p -\frac{1}{2}\beta_{st} x_s x_t + \sum_{s=1}^p \alpha_s x_s + \sum_{s=1}^p \sum_{j=1}^q \rho_{sj}(y_j) x_s \\ &+ \sum_{j=1}^q \sum_{r=1}^j \phi_{rj}(y_r, y_j) - \log Z(\Theta)\end{aligned} \tag{6}$$

The negative log-likelihood is convex, so standard gradient-descent algorithms can be used for computing the maximum likelihood estimates. The difficulty is that in each gradient step we have to compute $\hat{T}(x,y) - E_{p(\Theta)}\left[T(x,y)\right]$, the difference between the empirical sufficient statistic $\hat{T}(x,y)$ and the expected sufficient statistic. In both continuous and discrete graphical models the computationally expensive step is evaluating $E_{p(\Theta)}\left[T(x,y)\right]$. In discrete problems, this involves a sum over the discrete state space and in continuous problem, this requires matrix inversion. For both discrete and continuous models, there has been much work on on addressing these difficulties. For discrete models, the junction tree algorithm is an exact method for evaluating marginals and is suitable for models with low treewidth. Variational methods such as belief propagation and tree reweighted belief propagation work by optimizing a surrogate likelihood function by approximating the partition function $Z(\Theta)$ by a tractable surrogate $\widetilde{Z}(\Theta)$ (Wainwright and Jordan, 2008). In the case of a large discrete state space, these methods can be used to approximate $p(y)$ and do approximate maximum likelihood estimation for the discrete model. Approximate maximum likelihood estimation can also be done via monte carlo estimates of the gradients $\hat{T}(x,y) - E_{p(\Theta)}(T(x,y))$. For continuous Gaussian graphical models, efficient algorithms based on block coordinate descent (Friedman et al., 2008b; Banerjee et al., 2008) have been developed, that do not require matrix inversion. Since the pairwise mixed model includes both the discrete and continuous models as special cases, maximum likelihood estimation is at least as difficult as the two special cases. We defer the specifics of sampling from the model and maximum likelihood estimation to Appendices A and B.

3.1 Pseudolikelihood

The pseudolikelihood method Besag (1975) is a computationally efficient surrogate likelihood method formed by products of all the conditional distributions:

$$\tilde{\ell}(\Theta|x,y) = -\sum_{s=1}^{p} \log p(x_s|x_{\setminus s}, y;\Theta) - \sum_{r=1}^{q} \log p(y_r|x, y_{\setminus r};\Theta) \tag{7}$$

The conditional distributions $p(x_s|x_{\setminus s}, y;\theta)$ and $p(y_r = k|y_{\setminus r,}, x;\theta)$ take on the familiar form of linear gaussian and (multiclass) logistic regression.

- For a discrete variable y_r with L_r states, its conditional distribution is a multinomial distribution, as used in (multiclass) logistic regression. Whenever a discrete variable is a predictor, each level contributes an additive effect; continuous variables contribute linear effects.

  $$p(y_r|y_{\setminus r,}, x;\Theta) = \frac{\exp\left(\sum_s \rho_{sr}(y_r)x_s + \phi_{rr}(y_r,y_r) + \sum_{j\neq r}\phi_{rj}(y_r,y_j)\right)}{\sum_{l=1}^{L_r}\exp\left(\sum_s \rho_{sr}(l)x_s + \phi_{rr}(l,l) + \sum_{j\neq r}\phi_{rj}(l,y_j)\right)}$$

  This can be expressed in the more familiar form as

  $$p(y_r = k) = \frac{\exp\left(\alpha_k^T z\right)}{\sum_{l=1}^{L_r}\exp\left(\alpha_l^T z\right)} = \frac{\exp\left(\alpha_{0k} + \sum_j \alpha_{kj} z_j\right)}{\sum_{l=1}^{L_r}\exp\left(\alpha_{0l} + \sum_j \alpha_{lj} z_j\right)}$$

  The discrete variables are represented as dummy variables for each state $z_j = \mathbb{1}[y_u = k]$.

- Continuous variable x_s given all other variables is a gaussian distribution with a linear regression model for the mean.

  $$p(x_s|x_{\setminus s}, y;\Theta) = \frac{\sqrt{\beta_{ss}}}{\sqrt{2\pi}}\exp\left(\frac{-\beta_{ss}}{2}\left(\frac{\alpha_s + \sum_j \rho_{sj}(y_j) - \sum_{t\neq s}\beta_{st}x_t}{\beta_{ss}} - x_s\right)^2\right)$$

This can be expressed as

$$E(x_s|z_1,\ldots,z_p) = \alpha^T z = \alpha_0 + \sum_j z_j \alpha_j \tag{8}$$

$$p(x_s|z_1,\ldots,z_p) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x_s - \alpha^T z)^2\right) \text{ with } \sigma = 1/\beta_{ss} \tag{9}$$

Taking the negative log of both gives us

$$-\log p(x_s|x_{\backslash s}, y; \Theta) = -\frac{1}{2}\log \beta_{ss} + \frac{\beta_{ss}}{2}\left(\sum_j \frac{\rho_{sj}(y_j)}{\beta_{ss}} - \sum_{t\neq s} \frac{\beta_{st}}{\beta_{ss}} x_t - x_s\right)^2 \tag{10}$$

$$-\log p(y_r|y_{\backslash r,}, x; \Theta) = -\log \frac{\exp\left(\sum_s \rho_{sr}(y_r)x_s + \phi_{rr}(y_r,y_r) + \sum_{j\neq r}\phi_{rj}(y_r,y_j)\right)}{\sum_{l=1}^{L_r} \exp\left(\sum_s \rho_{sr}(l)x_s + \phi_{rr}(l,l) + \sum_{j\neq r}\phi_{rj}(l,y_j)\right)} \tag{11}$$

The discrete conditional $-\log p(y_r|y_{\backslash r,}, x; \Theta)$ is a standard (multiclass) logistic regression and thus convex in ρ_{sr} and ϕ_{rj}. The continuous conditional $-\log p(x_s|x_{\backslash s}, y; \Theta)$ is gaussian linear regression with unknown variance. This function is convex in β_{st}, $\rho_{sj}(y_j)$, and β_{ss} because x^2/y is jointly convex in x and $y > 0$. Thus the terms β_{st}^2/β_{ss} and ρ_{sj}^2/β_{ss} are convex. From this we can see that the negative log pseudolikelihood $\tilde{\ell}(\Theta)$ in (7) is jointly convex in Θ.

4. Conditional Independence and Penalty Terms

In this section, we show how to incorporate edge selection into the maximum likelihood or pseudolikelihood procedures. In the graphical representation of probability distributions, the absence of an edge $e = (u, v)$ corresponds to a conditional independency statement that variables x_u and x_v are conditionally independent given all other variables (Koller and Friedman, 2009). We would like to maximize the likelihood subject to a penalization on the number of edges since this results in a sparse graphical model. In the pairwise mixed model, there are 3 type of edges

1. β_{st} is a scalar that corresponds to an edge from x_s to x_t. $\beta_{st} = 0$ implies x_s and x_t are conditionally independent given all other variables. This parameter is in two conditional distributions, corresponding to either x_s or x_t is the response variable, $p(x_s|x_{\backslash s}, y; \Theta)$ and $p(x_t|x_{\backslash t}, y; \Theta)$.

2. ρ_{sj} is a vector of length L_j. If $\rho_{sj}(y_j) = 0$ for all values of y_j, then y_j and x_s are conditionally independent given all other variables. This parameter is in two conditional distributions, corresponding to either x_s or y_j is the response variable, $p(x_s|x_{\backslash s}, y; \Theta)$ and $p(y_j|x, y_{\backslash j}; \Theta)$.

3. ϕ_{rj} is a matrix of size $L_r \times L_j$. If $\phi_{rj}(y_r, y_j) = 0$ for all values of y_r and y_j, then y_r and y_j are conditionally independent given all other variables. This parameter is in two conditional distributions, corresponding to either y_r or y_j is the response variable, $p(y_r|x, y_{\backslash r}; \Theta)$ and $p(y_j|x, y_{\backslash j}; \Theta)$.

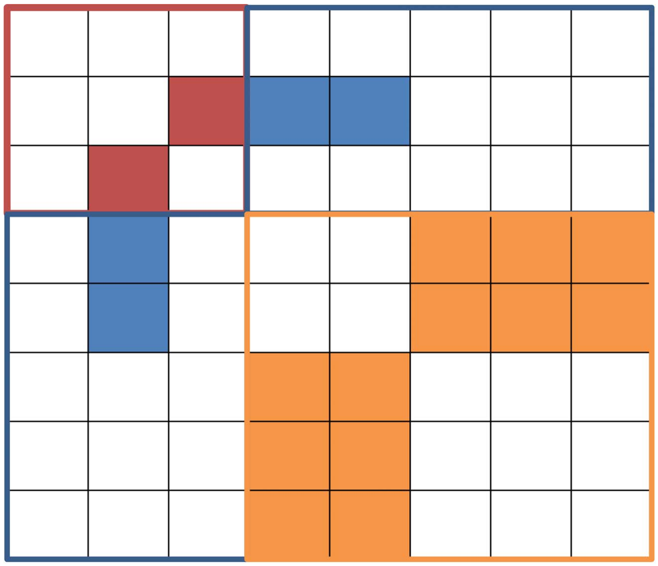


Figure 1: Symmetric matrix represents the parameters Θ of the model. This example has $p = 3$, $q = 2$, $L_1 = 2$ and $L_2 = 3$. The red square corresponds to the continuous graphical model coefficients B and the solid red square is the scalar β_{st}. The blue square corresponds to the coefficients ρ_{sj} and the solid blue square is a vector of parameters $\rho_{sj}(\cdot)$. The orange square corresponds to the coefficients ϕ_{rj} and the solid orange square is a matrix of parameters $\phi_{rj}(\cdot, \cdot)$.

For edges that involve discrete variables, the absence of that edge requires that the entire matrix ϕ_{rj} or vector ρ_{sj} is 0. This motivates the following regularlized optimization problem

$$\text{minimize}_{\Theta} \ell_\lambda(\Theta) = \ell(\Theta) + \lambda \left(\sum_{s<t} \mathbb{1}[\beta_{st} \neq 0] + \sum_{sj} \mathbb{1}[\rho_{sj} \not\equiv 0] + \sum_{r<j} \mathbb{1}[\phi_{rj} \not\equiv 0] \right) \tag{12}$$

All parameters that correspond to the same edge are grouped in the same indicator function. This problem is non-convex, so we replace the l_0 sparsity and group sparsity penalties with the appropriate convex relaxations. For scalars, we use the absolute value (l_1 norm), for vectors we use the l_2 norm, and for matrices we use the Frobenius norm. This choice corresponds to the standard relaxation from group l_0 to group l_1/l_2 (group lasso) norm (Bach et al., 2011; Yuan and Lin, 2006).

$$\text{minimize}_{\Theta} \ell_\lambda(\Theta) = \ell(\Theta) + \lambda \left(\sum_{t=1}^{p} \sum_{t=1}^{s-1} |\beta_{st}| + \sum_{s=1}^{p} \sum_{j=1}^{q} \|\rho_{sj}\|_2 + \sum_{j=1}^{q} \sum_{r=1}^{j-1} \|\phi_{rj}\|_F \right) \tag{13}$$

5. Calibrated regularizers

In this section, we consider calibrating the regularization penalties for each set of parameters. We introduce weights for each group of parameters and show how to choose the weights such that each parameter set is treated equally under the fully-factorized independence model [2]

$$\text{minimize}_{\Theta}\ell_\lambda(\Theta) = \ell(\Theta) + \lambda \left(\sum_{t=1}^{p} \sum_{t=1}^{s-1} w_{st}|\beta_{st}| + \sum_{s=1}^{p}\sum_{j=1}^{q} w_{sj}\left\|\rho_{sj}\right\|_2 + \sum_{j=1}^{q}\sum_{r=1}^{j-1} w_{rj}\left\|\phi_{rj}\right\|_F \right) \tag{14}$$

Based on the KKT conditions, the parameter group is non-zero if

$$\left\| \frac{\partial \ell}{\partial \theta_g} \right\| > \lambda w_g$$

where θ_g and w_g represents one of the parameter groups and its corresponding weight. Thus for all parameters to be on equal footing, we would like to choose the weights w such that

$$E_{p_F}\left\| \frac{\partial \ell}{\partial \theta_g} \right\| = \text{constant} \times w_g$$

However, it is simpler to compute in closed form $E_{p_F}\left\| \frac{\partial \ell}{\partial \theta_g} \right\|^2$, so we choose

$$w_g = \sqrt{E_{p_F}\left\| \frac{\partial \ell}{\partial \theta_g} \right\|^2}$$

In Appendix C, we show that the weights can be chosen as

$$\begin{aligned} w_{st} &= \sigma_s\sigma_t \\ w_{sj} &= \sigma_s\sqrt{\sum_a p_a(1-p_a)} \\ w_{rj} &= \sqrt{\sum_a p_a(1-p_a)\sum_b \hat{q}_b(1-\hat{q}_b)} \end{aligned}$$

σ_s is the standard deviation of the continuous variable x_s. $p_a = Pr(y_j = a)$. For all 3 types of parameters, the weight has the form of $w_{uv} = \mathbf{tr}(\mathbf{cov}(z_u))\mathbf{tr}(\mathbf{cov}(z_v))$, where z represents a generic variable and $\mathbf{cov}(z)$ is the variance-covariance matrix of z.

6. Optimization Algorithms

In this section, we discuss two algorithms for solving (13). This is a convex optimization problem that decomposes into the form $f(x) + g(x)$, where f is smooth and convex and g is convex but possibly non-smooth. In our case f is the negative log-likelihood or negative log-pseudolikelihood and g are the group sparsity penalties.

Block coordinate descent is a frequently used method when the non-smooth function g is the l_1 or group l_1. It is especially easy to apply when the function f is quadratic,

[2] Under the independence model p_F is fully-factorized $p(x,y) = \prod_{s=1}^{p} p(x_s)\prod_{r=1}^{q} p(y_r)$

since each block coordinate update can be solved in closed form for many different non-smooth g (Friedman et al., 2007). The smooth f in our particular case is not quadratic, so each block update cannot be solved in closed form. However in certain problems (sparse inverse covariance), the update can be approximately solved by using an appropriate inner optimization routine (Friedman et al., 2008b).

6.1 Proximal Gradient

Problems of this form are well-suited for the proximal gradient and accelerated proximal gradient algorithms as long as the proximal operator of g can be computed (Combettes and Pesquet, 2011; Beck and Teboulle, 2010)

$$prox_t(x) = argmin_u \frac{1}{2t} \|x-u\|^2 + g(u) \tag{15}$$

For the sum of l_2 group sparsity penalties considered, the proximal operator takes the familiar form of soft-thresholding and group soft-thresholding. Since the groups are non-overlapping, the proximal operator simplifies to scalar soft-thresholding for β_{st} and group soft-thresholding for ρ_{sj} and ϕ_{rj}.

The class of proximal gradient and accelerated proximal gradient algorithms is directly applicable to our problem. These algorithms work by solving a first-order model at the current iterate x_k

$$\begin{aligned} &\text{argmin}_u f(x_k) + \nabla f(x_k)^T(u-x_k) + \frac{1}{2t}\|u-x_k\|^2 + g(u) && (16)\\ &= \text{argmin}_u \frac{1}{2t}\|u-(x_k - t\nabla f(x_k))\|^2 + g(u) && (17)\\ &= prox_t(x_k - t\nabla f(x_k)) && (18)\\ &\text{The iteration is given by} && (19)\\ &x_{k+1} = prox_t\left(x_k - t\nabla f(x_k)\right) \text{ where } t \text{ is determined by line search} && (20) \end{aligned}$$

The TFOCS framework (Becker et al., 2011) is a package that allows us to experiment with 6 different variants of the accelerated proximal gradient algorithm. The TFOCS authors found that the Auslender-Teboulle algorithm exhibited less oscillatory behavior, and proximal gradient experiments in the next section were done using the Auslender-Teboulle implementation in TFOCS.

6.2 Proximal Newton Algorithms

This section borrows heavily from Schmidt (2010) and Schmidt et al. (2011). The class of proximal newton algorithms is a 2nd order analog of the proximal gradient algorithms. It attempts to incorporate 2nd order information about the smooth function f into the model function. At each iteration, it minimizes a quadratic model centered at x_k

$$\text{argmin}_u f(x_k) + \nabla f(x_k)^T(u - x_k) + \frac{1}{2t}(u - x_k)^T H(u - x_k) + g(u) \tag{21}$$

$$= \text{argmin}_u \frac{1}{2t}\left(u - x_k + tH^{-1}\nabla f(x_k)\right)^T H\left(u - x_k + tH^{-1}\nabla f(x_k)\right) + g(u) \tag{22}$$

$$= argmin_u \frac{1}{2t}\left\|u - \left(x_k - tH^{-1}\nabla f(x_k)\right)\right\|_H^2 + g(u) \tag{23}$$

$$:= Hprox_t\left(x_k - tH^{-1}\nabla f(x_k)\right) \text{ where } H = \nabla^2 f(x_k) \tag{24}$$

The $Hprox$ operator is analogous to the proximal operator, but in the $\|\cdot\|_H$-norm. It simplifies to the proximal operator if $H = I$, but in the general case of positive definite H there is no closed-form solution for many common non-smooth $g(x)$ (including l_1 and group l_1). However if the proximal operator of g is available, each of these sub-problems can be solved efficiently with proximal gradient. In the case of separable g, coordinate descent is also applicable. Fast methods for solving the subproblem $Hprox_t(x_k - tH^{-1}\nabla f(x_k))$ include coordinate descent methods, proximal gradient methods, or Barzilai-Borwein (Friedman et al., 2007; Combettes and Pesquet, 2011; Beck and Teboulle, 2010; Wright et al., 2009). The proximal Newton framework allows us to bootstrap many previously developed solvers to the case of arbitrary loss function f.

Algorithm 1 Proximal Newton

```
repeat
    Solve subproblem $p_k = Hprox_t\left(x_k - tH_k^{-1}\nabla f(x_k)\right) - x_k$ using TFOCS.
    Find $t$ to satisfy Armijo line search condition with parameter $\alpha$
        $f(x_k + tp_k) + g(x_k + tp_k) \le f(x_k) + g(x_k) - \frac{t\alpha}{2}\|p_k\|^2$
    Set $x_{k+1} = x_k + tp_k$
    $k = k + 1$
until $\frac{\|x_k - x_{k+1}\|}{\|x_k\|} < tol$
```

Proximal Newton methods generally require fewer outer iterations (evaluations of $Hprox$) than first-order methods while providing higher accuracy because they incorporate 2nd order information. They are also widely applicable since the algorithm only needs access to the ∇f and proximal operator of g, if each subproblem is solved using proximal gradient or Barzilai Borwein. Since the objective is quadratic, coordinate descent is also applicable to the subproblems. The hessian matrix H can be replaced by a quasi-newton approximation such as BFGS/L-BFGS/SR1. In our implementation, we use a L-BFGS approximation and solve the subproblem with TFOCS.

6.3 Path Algorithm

Frequently in machine learning and statistics, the regularization parameter λ is heavily dependent on the dataset. λ is generally chosen via cross-validation or holdout set performance, so it is convenient to provide solutions over an interval of $[\lambda_{min}, \lambda_{max}]$. We start the algorithm at $\lambda_1 = \lambda_{max}$ and solve, using the previous solution as warm start, for $\lambda_2 > \ldots > \lambda_{min}$. We find that this reduces the cost of fitting an entire path of solutions (See Figure 3). λ_{max} can be chosen as the smallest value such that all parameters are 0 by using the subgradient equations.

7. Conditional Model

We can generalize our mixed model to include a conditional model by incorporating features, similar to a conditional random field. Conditional models only model the conditional distribution $p(z|f)$, as opposed to the joint distribution $p(z, f)$, where z are the variables of interest to the prediction task and f are features.

In addition to observing x and y, we observe features f and we build a graphical model for the conditional distribution $p(x, y|f)$. Consider a full pairwise model $p(x, y, f)$ of the form (1). We then choose to only model the joint distribution over only the variables x and y to give us $p(x, y|f)$ which is of the form

$$\begin{aligned} p(x,y|f;\Theta) = \frac{1}{Z(\Theta|f)} \exp \Bigg(& \sum_{s=1}^{p}\sum_{t=1}^{p} -\frac{1}{2}\beta_{st}x_s x_t + \sum_{s=1}^{p}\alpha_s x_s + \sum_{s=1}^{p}\sum_{j=1}^{q}\rho_{sj}(y_j)x_s \\ & + \sum_{j=1}^{q}\sum_{r=1}^{j}\phi_{rj}(y_r,y_j) + \sum_{l=1}^{F}\sum_{s=1}^{p}\gamma_{ls}x_s f_l + \sum_{l=1}^{F}\sum_{r=1}^{q}\eta_{lr}(y_r) f_l \Bigg) \end{aligned} \tag{25}$$

We can also consider a more general model where each pairwise edge potential depends on the features

$$\begin{aligned} p(x,y|f;\Theta) = \frac{1}{Z(\Theta|f)} \exp \Bigg(& \sum_{s=1}^{p}\sum_{t=1}^{p} -\frac{1}{2}\beta_{st}(f)x_s x_t + \sum_{s=1}^{p}\alpha_s(f) x_s \\ & + \sum_{s=1}^{p}\sum_{j=1}^{q}\rho_{sj}(y_j,f)x_s + \sum_{j=1}^{q}\sum_{r=1}^{j}\phi_{rj}(y_r,y_j,f) \Bigg) \end{aligned} \tag{26}$$

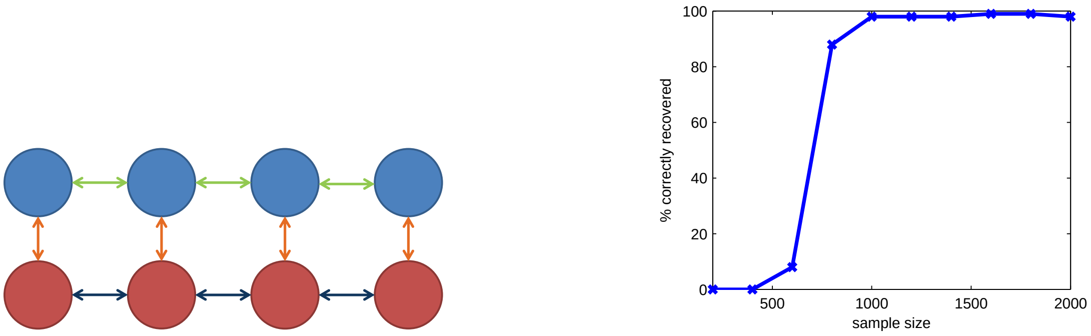
(25) is a special case of this where only the node potentials depend on features and the pairwise potentials are independent of feature values. The specific parametrized form we consider is $\phi_{rj}(y_r, y_j, f) \equiv \phi_{rj}(y_r, y_j)$ for $r \neq j$, $\rho_{sj}(y_j, f) \equiv \rho_{sj}(y_j)$, and $\beta_{st}(f) = \beta_{st}$. The node potentials depend linearly on the feature values, $\alpha_s(f) = \alpha_s + \sum_{l=1}^{F} \gamma_{ls} x_s f_l$, and $\phi_{rr}(y_r, y_r, f) = \phi_{rr}(y_r, y_r) + \sum_l \eta_{lr}(y_r)$.

8. Experimental Results

We present experimental results on synthetic data, survey data and on a conditional model.

8.1 Synthetic Experiments

In the synthetic experiment, the training points are sampled from a true model with 10 continuous variables and 10 binary variables. The edge structure is shown in Figure 2a. λ



(a) True Graph Structure

(b) Probability of recovery

Figure 2: Figure 2a shows the graph used in the synthetic experiments for $p = q = 4$. Blue nodes are continuous variables, red nodes are binary variables and the orange, green and dark blue lines represent the 3 types of edges. Figure 2b is a plot of the probability of correct edge recovery at a given sample size. Results are averaged over 100 trials.

is chosen as $5\sqrt{\frac{\log p+q}{n}}$ as suggested in several theoretical studies (Ravikumar et al., 2010; Jalali et al., 2011). We see from the experimental results that recovery of the correct edge set undergoes a sharp phase transition, as expected. With $n = 1000$ samples, we are recovering the correct edge set with probability nearly 1. This is expected since Meinshausen and Bühlmann (2006) showed that for sparse gaussian graphical models node-wise linear regression recovers the true edge set. Similarly for discrete graphical models, Ravikumar et al. (2010) and Jalali et al. (2011) showed respectively that logistic regression, and (multiclass) logistic regression recover the true edge set under certain incoherence assumptions. The phase transition experiments were done using the proximal Newton algorithm discussed in Section 6.2.

8.2 Survey Experiments

The survey dataset we consider consists of 11 variables, of which 2 are continuous and 9 are discrete: age (continuous), log-wage (continuous), year(7 states), sex(2 states),marital status (5 states), race(4 states), education level (5 states), geographic region(9 states), job class (2 states), health (2 states), and health insurance (2 states). All the evaluations are done using a holdout test set of size $100,000$ for the survey experiments. The regularization parameter λ is varied over the interval $[5 \times 10^{-5}, .7]$ at 50 points equispaced on log-scale for all experiments.

8.2.1 Model Selection

In Figure 3, we study the model selection performance of learning a graphical model over the 11 variables under different training samples sizes. We see that as the sample size increases, the optimal model is increasingly dense, and less regularization is needed.

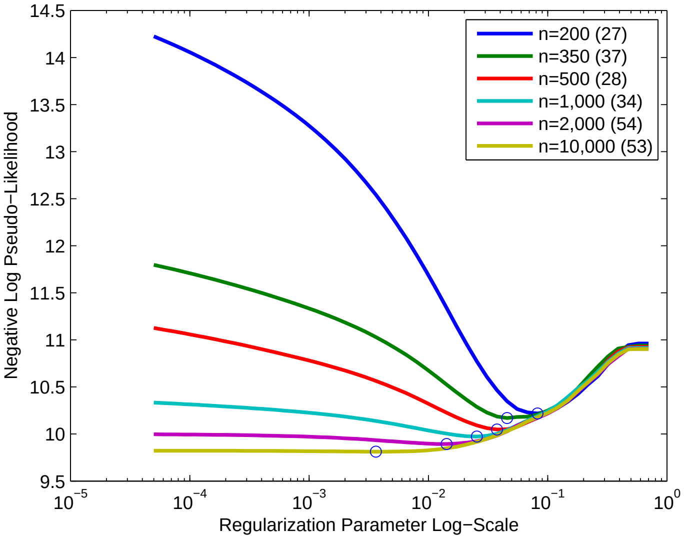


Figure 3: Model selection under different training set sizes. Circle denotes the lowest test set negative log pseudolikelihood and the number in parentheses is the number of edges in that model at the lowest test negative log pseudolikelihood. The saturated model has 55 edges.

8.2.2 Comparing against Separate Regressions

A sensible baseline method to compare against is a separate regression algorithm. This algorithm fits a linear gaussian or (multiclass) logistic regression of each variable conditioned on the rest. We can evaluate the performance of the pseudolikelihood by evaluating $-\log p(x_s|x_{\backslash s}, y)$ for linear regression and $-\log p(y_r|y_{\backslash r}, x)$ for (multiclass) logistic regression. Since regression is directly optimizing this loss function, it is expected to do better. The pseudolikelihood objective is similar, but has half the number of parameters as the separate regressions since the coefficients are shared between two of the conditional likelihoods. From Figures 4 and 5, we can see that the pseudolikelihood performs very similarly to the separate regressions and sometimes even outperforms regression. The benefit of the pseudolikelihood is that we have learned parameters of the joint distribution $p(x, y)$ and not just of the conditionals $p(x_s|y, x_{\backslash s})$. On the test dataset, we can compute quantities such as conditionals over arbitrary sets of variables $p(y_A, x_B|y_{A^C}, x_{B^C})$ and marginals $p(x_A, y_B)$ (Koller and Friedman, 2009). This would not be possible using the separate regressions.

8.2.3 Conditional Model

Using the conditional model (25), we model only the 3 variables logwage, education(5) and jobclass(2). The other 8 variables are only used as features. The conditional model is then trained using the pseudolikelihood. We compare against the generative model that learns a joint distribution on all 11 variables. From Figure 6, we see that the conditional model

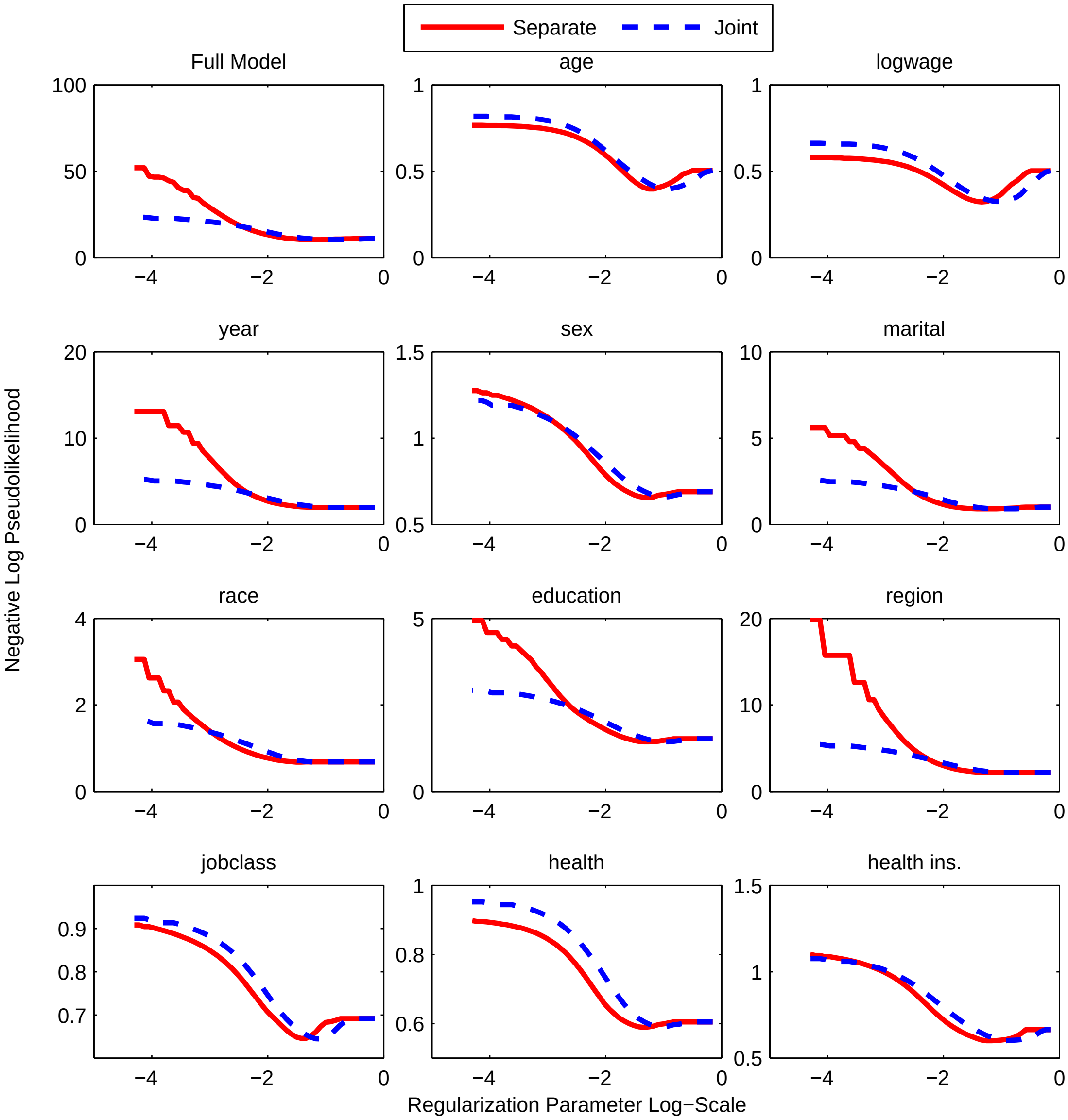


Figure 4: Separate Regression vs Pseudolikelihood $n = 100$. y-axis is the appropriate regression loss for the response variable. For low levels of regularization and at small training sizes, the pseudolikelihood seems to overfit less; this may be due to a global regularization effect from fitting the joint distribution as opposed to separate regressions.

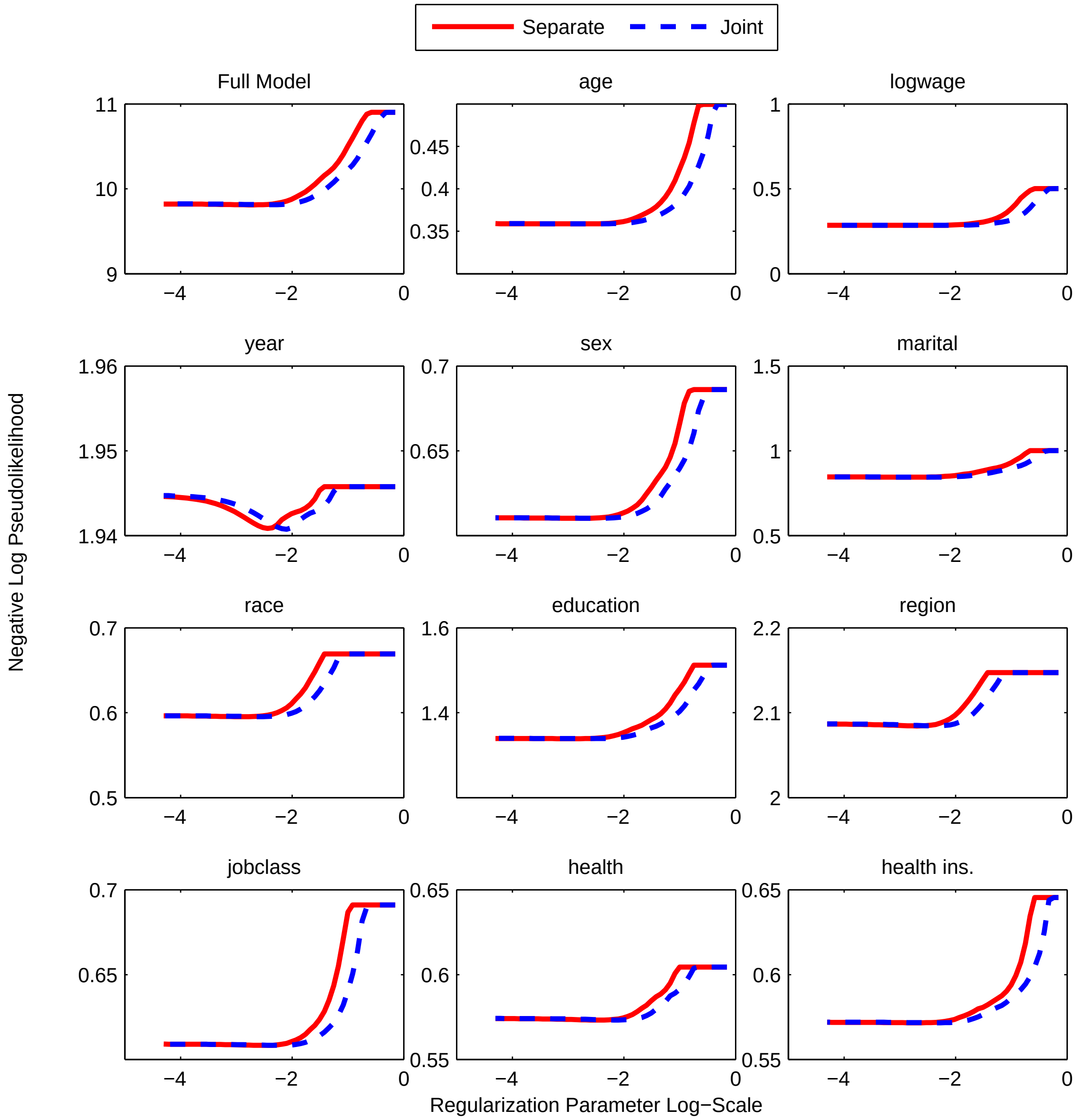


Figure 5: Separate Regression vs Pseudolikelihood $n = 10,000$. y-axis is the appropriate regression loss for the response variable. At large sample sizes, separate regressions and pseudolikelihood perform very similarly. This is expected since this is nearing the asymptotic regime.

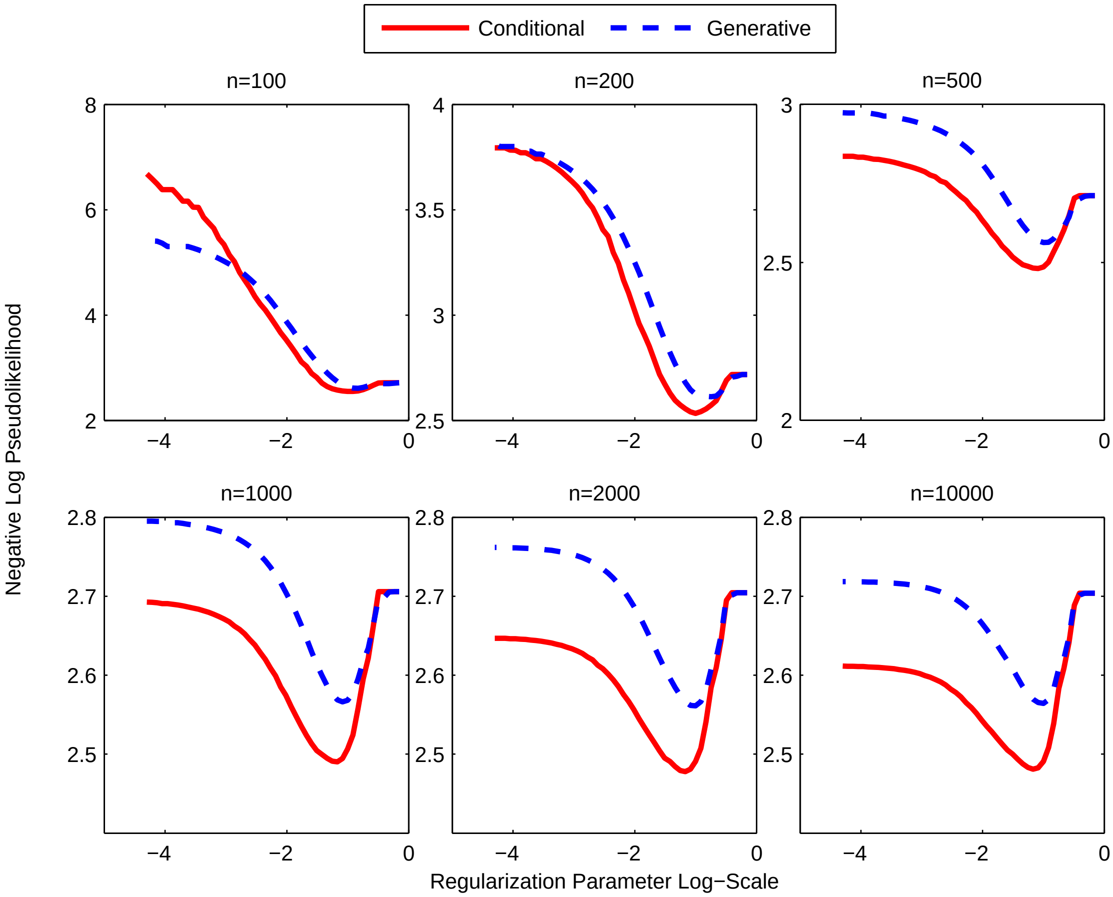


Figure 6: Conditional Model vs Generative Model at various sample sizes. y-axis is test set performance is evaluated on negative log pseudolikelihood of the conditional model. The conditional model outperforms the full generative model at except the smallest sample size $n = 100$.

outperforms the generative model, except at small sample sizes. This is expected since the conditional distribution models less variables. At very small sample sizes and small λ, the generative model outperforms the conditional model. This is likely because generative models converge faster (with less samples) than discriminative models to its optimum.

8.2.4 Maximum Likelihood vs Pseudolikelihood

The maximum likelihood estimates are computable for very small models such as the conditional model previously studied. The pseudolikelihood was originally motivated as an approximation to the likelihood that is computationally tractable. We compare the maximum likelihood and maximum pseudolikelihood on two different evaluation criteria: the negative log likelihood and negative log pseudolikelihood. In Figure 7, we find that the pseudolikelihood outperforms maximum likelihood under both the negative log likelihood and

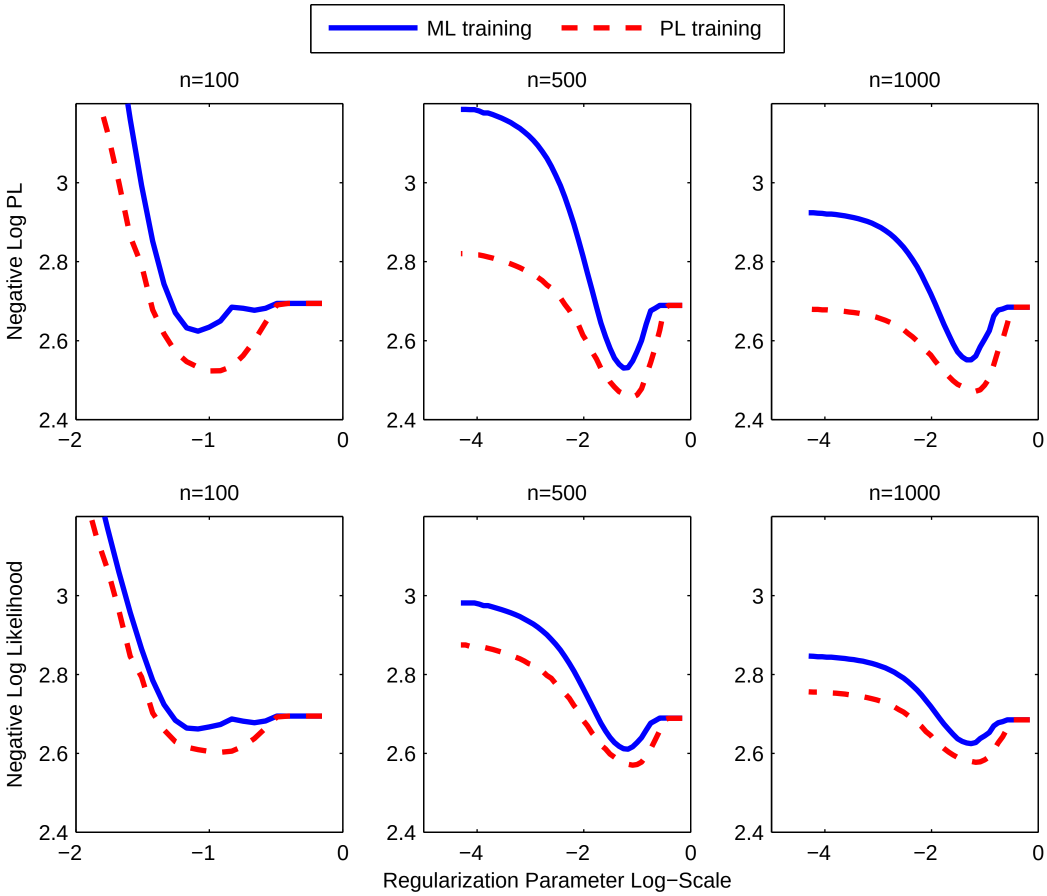


Figure 7: Maximum Likelihood vs Pseudolikelihood. y-axis for top row is the negative log pseudolikelihood. y-axis for bottom row is the negative log likelihood. Pseudolikelihood outperforms maximum likelihood across all the experiments.

negative log pseudolikelihood. We would expect that the pseudolikelihood trained model does better on the pseudolikelihood evaluation and maximum likelihood trained model does better on the likelihood evaluation. However, we found that the pseudolikelihood trained model outperformed the maximum likelihood trained model on both evaluation criteria. Although asymptotic theory suggests that maximum likelihood is more efficient than the pseudolikelihood, this analysis may not be applicable because of the finite sample regime and misspecified model (Liang and Jordan, 2008).

Acknowledgments

We would like to thank Yuekai Sun for providing software on the proximal Newton method and Percy Liang for helpful discussions. Jason Lee is supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship

(NDSEG) Program, National Science Foundation Graduate Research Fellowship Program, and the Stanford Graduate Fellowship. Trevor Hastie was partially supported by grant DMS-1007719 from the National Science Foundation, and grant RO1-EB001988-15 from the National Institutes of Health.

References

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4:1–106, 2011. URL `http://dx.doi.org/10.1561/2200000015`.

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery problems. *Convex Optimization in Signal Processing and Communications*, pages 42–88, 2010.

S.R. Becker, E.J. Candès, and M.C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, pages 1–54, 2011.

J. Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975.

P.L. Combettes and J.C. Pesquet. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212, 2011.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008a.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008b.

J. Friedman, T. Hastie, and R. Tibshirani. Applications of the lasso and grouped lasso to the estimation of sparse graphical models. Technical report, Technical Report, Stanford University, 2010.

H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *The Journal of Machine Learning Research*, 10:883–906, 2009.

A. Jalali, P. Ravikumar, V. Vasuki, S. Sanghavi, UT ECE, and UT CS. On learning discrete graphical models using group-sparse regularization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

S.L. Lauritzen. *Graphical models*, volume 17. Oxford University Press, USA, 1996.

S.I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of markov networks using l1regularization. In *NIPS*, 2006.

P. Liang and M.I. Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of the 25th international conference on Machine learning*, pages 584–591. ACM, 2008.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

J. Peng, P. Wang, N. Zhou, and J. Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

P. Ravikumar, M.J. Wainwright, and J.D. Lafferty. High-dimensional ising model selection using l1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

M. Schmidt. *Graphical Model Structure Learning with l1-Regularization.* PhD thesis, University of British Columbia, 2010.

M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. *CVPR. IEEE Computer Society*, 2008.

M. Schmidt, D. Kim, and S. Sra. Projected newton-type methods in machine learning. 2011.

M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

S.J. Wright, R.D. Nowak, and M.A.T. Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493, 2009.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Appendix A. Sampling From The Joint Distribution

In this section we discuss how to draw samples $(x, y) \sim p(x, y)$. Using the property that $p(x, y) = p(y)p(x|y)$, we see that if $y \sim p(y)$ and $x \sim p(x|y)$ then $(x, y) \sim p(x, y)$. We have that

$$p(y) \propto \exp\,(\sum_{r,j} \phi_{rj}(y_r, y_j) + \frac{1}{2}\rho(y)^T B^{-1} \rho(y)) \tag{27}$$

$$(\rho(y))_s = \sum_j \rho_{sj}(y_j) \tag{28}$$

$$p(x|y) = No(B^{-1}(\alpha + \rho(y)), B^{-1}) \tag{29}$$

The difficult part is to sample $y \sim p(y)$ since this involves the partition function of the discrete MRF. This can be done with MCMC for larger models and junction tree algorithm or exact sampling for small models.

Appendix B. Maximum Likelihood

The joint distribution and loglikelihood are:

$$\begin{aligned} p(x, y; \Theta) &= \exp\,(-\frac{1}{2}x^T Bx + (\alpha + \rho(y))^T x + \sum_{(r,j)} \phi_{rj}(y_r, y_j))/Z(\Theta) \\ \ell(\Theta) &= \left(\frac{1}{2}x^T Bx - (\alpha + \rho(y))^T x - \sum_{(r,j)} \phi_{rj}(y_r, y_j)\right) \\ &\quad + \log(\sum_{y'} \int dx \exp\,(-\frac{1}{2}x^T Bx + (\alpha + \rho(y'))^T x) \exp(\sum_{(r,j)} \phi_{rj}(y'_r, y'_j))) \end{aligned}$$

The derivative is

$$\begin{aligned} \frac{\partial \ell}{\partial B} &= \frac{1}{2}xx^T + \frac{\int dx(\sum_{y'} -\frac{1}{2}xx^T \exp(-\frac{1}{2}x^T Bx + (\alpha + \rho(y))^T x + \sum_{(r,j)} \phi_{rj}(y'_r, y'_j)))}{Z(\Theta)} \\ &= \frac{1}{2}xx^T + \int \sum_{y'} (-\frac{1}{2}xx^T p(x, y'; \Theta)) \\ &= \frac{1}{2}xx^T + \sum_{y'} \int -\frac{1}{2}xx^T p(x|y'; \Theta)p(y') \\ &= \frac{1}{2}xx^T + \sum_{y'} \int -\frac{1}{2} \left(B^{-1} + B^{-1}(\alpha + \rho(y'))(\alpha + \rho(y')^T)B^{-1}\right) p(y') \end{aligned}$$

The primary cost is to compute B^{-1} and the sum over the discrete states y.
The computation for the derivatives of $\ell(\Theta)$ with respect to ρ_{sj} and ϕ_{rj} are similar.

$$\begin{aligned}\frac{\partial \ell}{\phi_{rj}(a,b)} &= -1(y_r=a, y_j=b) + \sum_{y'} \int dx 1(y'_r=a, y'_j=b) p(x,y';\Theta) \\ &= -1(y_r=a, y_j=b) + \sum_{y'} 1(y'_r=a, y'_j=b) p(y')\end{aligned}$$

The gradient requires summing over all discrete states.
Similarly for $\rho_{sj}(a)$:

$$\begin{aligned}\frac{\partial \ell}{\rho_{sj}(a)} &= -1(y_j=a)x_s + \sum_{y'} \int dx (1(y'_j=a)x_s) p(x',y';\Theta) \\ = -1(y_j=a)x_s &+ \int dx \sum_{y'_{\backslash j}} x_s p(x|y'_{\backslash j}, y'_j=a) p(y'_{\backslash j}, y'_j=a)\end{aligned}$$

MLE estimation requires summing over the discrete states to compute the expected sufficient statistics. This may be approximated using using samples $(x,y) \sim p(x,y;\Theta)$. The method in the previous section shows that sampling is efficient if $y \sim p(y)$ is efficient. This allows us to use fast MCMC methods developed for discrete MRF's such as the Swendsen-wang method.

Appendix C. Choosing the Weights

We first show how to compute w_{sj}. The gradient of the pseudo-likelihood with respect to a parameter $\rho_{sj}(a)$ is given below

$$\begin{aligned}\frac{\partial \tilde{\ell}}{\partial \rho_{sj}(a)} &= \sum_{i=1}^{n} -2 \times \mathbb{1}\left[y_j^i = a\right] x_s^i + E_{p_F}(\mathbb{1}[y_j=a]x_s | y_{\backslash j}^i, x^i) + E_{p_F}(\mathbb{1}[y_j=a]x_s | x_{\backslash s}^i, y^i) \\ &= \sum_{i=1}^{n} -2 \times \mathbb{1}\left[y_j^i = a\right] x_s^i + x_s^i p(y_j=a) + \mathbb{1}\left[y_j^i=a\right]\mu_s \\ &= \sum_{i=1}^{n} \mathbb{1}\left[y_j^i=a\right]\left(\hat{\mu}_s - x_s^i\right) + x_s^i\left(\hat{p}(y_j=a) - \mathbb{1}\left[y_j^i=a\right]\right) \\ &= \sum_{i=1}^{n} \left(\mathbb{1}\left[y_j^i=a\right] - \hat{p}(y_j=a)\right)\left(\hat{\mu}_s - x_s^i\right) + \left(x_s^i - \hat{\mu}_s\right)\left(\hat{p}(y_j=a) - \mathbb{1}\left[y_j^i=a\right]\right) \qquad (30) \\ &= \sum_{i=1}^{n} 2\left(\mathbb{1}\left[y_j^i=a\right] - \hat{p}(y_j=a)\right)\left(\hat{\mu}_s - x_s^i\right) \qquad (31)\end{aligned}$$

Since the subgradient condition includes a variable if $\left\|\frac{\partial\tilde{\ell}}{\partial\rho_{sj}}\right\| > \lambda$, we compute $E\left\|\frac{\partial\tilde{\ell}}{\partial\rho_{sj}}\right\|^2$. By independence,

$$\begin{aligned}
&E_{p_F}\left(\left\|\sum_{i=1}^n 2\left(\mathbb{1}\left[y_j^i = a\right] - \hat{p}(y_j = a)\right)\left(\hat{\mu}_s - x_s^i\right)\right\|^2\right) && (32)\\
&= 4nE_{p_F}\left(\left\|\mathbb{1}\left[y_j^i = a\right] - \hat{p}(y_j = a)\right\|^2\right)E_{p_F}\left(\left\|\hat{\mu}_s - x_s^i\right\|^2\right) && (33)\\
&= 4(n-1)p(y_j = a)(1 - p(y_j = a))\sigma_s^2 && (34)
\end{aligned}$$

The last line is an equality if we replace the sample means $\hat{p}$ and $\hat{\mu}$ with the true values p and μ. Thus for the entire vector ρ_{sj} we have $E_{p_F}\left\|\frac{\partial\tilde{\ell}}{\partial\rho_{sj}}\right\|^2 = 4(n-1)\left(\sum_a p(y_j = a)(1 - p(y_j = a)\right)\sigma_s^2$. If we let the vector z be the indicator vector of the categorical variable y_j, and let the vector $p = p(y_j = a)$, then $E_{p_F}\left\|\frac{\partial\tilde{\ell}}{\partial\rho_{sj}}\right\|^2 = 4(n-1)\sum_a p_a(1-p_a)\sigma^2 = 4(n-1)\mathbf{tr}(\mathbf{cov}(z))\mathbf{var}(x)$ and $w_{sj} = \sqrt{\sum_a p_a(1-p_a)\sigma_s^2}$.

We repeat the computation for β_{st}.

$$\begin{aligned}
\frac{\partial\ell}{\partial\beta_{st}} &= \sum_{i=1}^n -2x_s^i x_t + E_{p_F}(x_s^i x_t^i | x_{\backslash s}, y) + E_{p_F}(x_s^i x_t^i | x_{\backslash t}, y)\\
&= \sum_{i=1}^n -2x_s^i x_t^i + \hat{\mu}_s x_t^i + \hat{\mu}_t x_s^i\\
&= \sum_{i=1}^n x_t^i(\hat{\mu_s} - x_s^i) + x_s^i(\hat{\mu}_t - x_t^i)\\
&= \sum_{i=1}^n (x_t^i - \hat{\mu}_t)(\hat{\mu_s} - x_s^i) + (x_s^i - \hat{\mu_s})(\hat{\mu}_t - x_t^i)\\
&= \sum_{i=1}^n 2(x_t^i - \hat{\mu}_t)(\hat{\mu_s} - x_s^i)
\end{aligned}$$

Thus

$$\begin{aligned}
&E\left(\left\|\sum_{i=1}^n 2(x_t^i - \hat{\mu}_t)(\hat{\mu_s} - x_s^i)\right\|^2\right)\\
&= 4nE_{p_F}\left\|x_t - \hat{\mu_t}\right\|^2 E_{p_F}\left\|x_s - \hat{mu}_s\right\|^2\\
&= 4(n-1)\sigma_s^2\sigma_t^2
\end{aligned}$$

Thus $E_{p_F} \left\| \frac{\partial \ell}{\partial \beta_{st}} \right\|^2 = 4(n-1)\sigma_s^2 \sigma_t^2$ and taking square-roots gives us $w_{st} = \sigma_s \sigma_t$. We repeat the same computation for ϕ_{rj}. Let $p_a = Pr(y_r = a)$ and $\hat{q}_b = Pr(y_j = b)$.

$$\begin{aligned}
\frac{\partial \tilde{\ell}}{\partial \phi_{rj}(a,b)} &= \sum_{i=1}^{n} -\mathbb{1}\left[y_r^i = a\right]\mathbb{1}\left[y_j^i = b\right] + E\left(\mathbb{1}[y_r = a]\mathbb{1}[y_j = b]|y_{\setminus r}, x\right) \\
&+ E\left(\mathbb{1}[y_r = a]\mathbb{1}[y_j = b]|y_{\setminus j}, x\right) \\
&= \sum_{i=1}^{n} -\mathbb{1}\left[y_r^i = a\right]\mathbb{1}\left[y_j^i = b\right] + \hat{p}_a \mathbb{1}\left[y_j^i = b\right] + \hat{q}_b \mathbb{1}\left[y_r^i = a\right] \\
&= \sum_{i=1}^{n} \mathbb{1}\left[y_j^i = b\right](\hat{p}_a - \mathbb{1}\left[y_r^i = a\right]) + \mathbb{1}\left[y_r^i = a\right](\hat{q}_b - \mathbb{1}\left[y_j^i = b\right]) \\
&= \sum_{i=1}^{n} (\mathbb{1}\left[y_j^i = b\right] - \hat{q}_b)(\hat{p}_a - \mathbb{1}\left[y_r^i = a\right]) + (\mathbb{1}\left[y_r^i = a\right] - \hat{p}_a)(\hat{q}_b - \mathbb{1}\left[y_j^i = b\right]) \\
&= \sum_{i=1}^{n} 2(\mathbb{1}\left[y_j^i = b\right] - \hat{q}_b)(\hat{p}_a - \mathbb{1}\left[y_r^i = a\right])
\end{aligned}$$

Thus we compute

$$\begin{aligned}
E_{p_F} \left\| \frac{\partial \tilde{\ell}}{\partial \phi_{rj}(a,b)} \right\|^2 &= E\left(\left\| \sum_{i=1}^{n} 2(\mathbb{1}\left[y_j^i = b\right] - \hat{q}_b)(\hat{p}_a - \mathbb{1}\left[y_r^i = a\right]) \right\|^2 \right) \\
&= 4n E_{p_F} \left\| \hat{q}_b - \mathbb{1}[y_j = b] \right\|^2 E_{p_F} \left\| \hat{p}_a - \mathbb{1}[y_r = a] \right\|^2 \\
&\approx 4(n-1) q_b(1-q_b) p_a(1-p_a)
\end{aligned}$$

From this, we see that $E_{p_F} \left\| \frac{\partial \tilde{\ell}}{\partial \phi_{rj}} \right\|^2 = \sum_{a=1}^{L_r} \sum_{b=1}^{L_j} 4(n-1) q_b(1-q_b) p_a(1-p_a)$ and $w_{rj} = \sqrt{\sum_{a=1}^{L_r} \sum_{b=1}^{L_j} q_b(1-q_b) p_a(1-p_a)}$